Hessian Free Optimization

ritchie huang

2019-09-09

Contents

1 Background

we can get approximation of f(x) in second-order Tylor expansion:

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \Delta x^T H(f) \Delta x \tag{1}$$

2 Conjugate Gradient

Suppose we define function as bellow:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c.\ x, b, c \in \mathbb{R}^n \tag{2}$$

where $A^T = A$, that's say A is symmetric.

2.1 Prerequisite

Taking gradient of f, we obtain:

$$\nabla f(x) = Ax + b \tag{3}$$

the direction of gradient is the direction in which the function rises fatest.

If we hope to minimize our function, we can then update x by:

$$x = x - \alpha \nabla f(x) \tag{4}$$

where α is called step-size (may be learning rate), we perform a line search to find the best α when the direction $d_0 = -\nabla f(x)$ is given.

Note that choosing the best α is equivalent to minimize the following function:

$$\begin{aligned} g(\alpha) &= f(x_0 + \alpha d_0) \\ &= \frac{1}{2}(x_0 + \alpha d_0)^T A (x_0 + \alpha d_0) + b^T (x_0 + \alpha d_0) + c \\ &= \frac{1}{2} \alpha^2 d_0^T A d_0 + d_0^T (A x_0 + b) \alpha + \left(\frac{1}{2} x_0^T A x_0 + b^T x_0 + c \right) \end{aligned} \tag{5}$$

More generally, when we update x_i iteratively, since $g(\alpha)$ is a quadratic function in α, it has a unique global minimum or maximum.Assume this function has a global minimum where $g'(\alpha) = 0$:

$$g'(\alpha) = \alpha (d_i^T A d_i) + d_i^T (A x_i + b) = 0 \tag{6}$$

Solving the equation above, we obtain:

$$\alpha = -\frac{d_i^T(Ax_i + b)}{d_i^T A d_i} \tag{7}$$

Thus, we can update x_1 using x_0 and α in the iterative algorithm:

$$x_1 = x_0 + \alpha \nabla f(x_0) \tag{8}$$

However, it's subtle that each α_i is related to $d_i = -\nabla f(x_i)$, when we are going to update x_{i+1}, we may ruin our update from previous iteration.Therefore, we need to rectify direction d_{i+1}, which is conjugate to d_i.

$$d_{i+1} = -\nabla f(x_{i+1}) + \beta_i d_i \tag{9}$$

But, what's β_i ? We can derive it from the conjugacy between d_{i+1} and d_i.

We define vector x and y to be conjugate w.r.t a semi-definite matrix A if $x^T Ay = 0$.

We obtain that:

$$\begin{aligned} d_{i+1}^T A d_i &= 0 \\ &= \left(-\nabla f(x_{i+1}) + \beta_i d_i\right)^T A d_i \\ &= -\nabla f(x_{i+1})^T A d_i + \beta_i d_i^T A d_i \end{aligned} \tag{10}$$

Solve the equation above:

$$\beta_i = \frac{\nabla f(x_{i+1})^T A d_i}{d_i^T A d_i} \tag{11}$$

Combining all steps above, let's writing the pseudocode for Conjugate Gradient Alorithm in the next section.

3 Pseudocode for Conjugate Gradient

Give function : $f(x)$, the algorithm is going to update x to minimize : $f(x)$ numerically and iteratively. First, we approximate this function to twice Tylor Expansion:

$$f(x + \Delta x) \approx f(x) + b^T \Delta x + \Delta x^T A \Delta x \tag{12}$$

Algorithm 1 Conjugate Gradient Algorithm

Input: A, b, n

Output: x

Initialize **k** $= 0$, $x_k = 0$, $d_k = -(Ax_k + b)$.

while k < n **do**

  Select the best step size $\alpha = -\frac{d_k^T d_k}{d_k^T A d_k}$

  Update solution $x_{k+1} = x_k + \alpha d_k$

  $d_{k+1} = Ax_k + b$

  $\beta = \frac{d_{k+1}^T A d_k}{d_k^T A d_k}$

  Rectify update direction $d_{k+1} = d_{k+1} + \beta d_k$

  $k \leftarrow k + 1$

return x